Overview

Photon mapping can be used to simulate the refraction of light through transparent objects like water or glass. This allows to produce effects like Caustics, Indirect Illumination and Volumetric Scattering of light, which cannot be generated by conventional ray tracing . The photon mapping

algorithm can be described in following steps.

1. **Emitting Photons** - A large number of photons are thrown in the scene by every light source. Total light energy in the scene will not depend on number of photons.
2. **Tracing photons** - Tracing of photons is similar to that of viewing rays. They will be reflected, refracted and absorbed in the scene depending on material properties.
3. **Storing Photons** - Position, color and direction of photons in the scene have to be stored. A good data structure is necessary as it is important to locate photons in the neighborhood of the ray-intersection. Typically, A kd-tree is used for faster implementation.
4. **Visualizing Photons** - For shading equation, the density of the photon-map has to be visualized. We use the point of intersection and search for photons closest to it. The color components are modified depending on the average color of the photons.

Dataset:

We will use the framework similar to the one shared in class for Ray tracing project, and extend it to implement photon mapping.

What is unique:

As compared to traditional ray tracing, where we only throw viewing rays into the scene, photon mapping uses a two-pass rendering approach. In the first pass, photons are traced from light sources into the scene and get deposited after having interacted with objects in the scene. In the second pass, viewing rays are traced into the scene and the shading color of an intersection is determined based on the number and color of the photons in its neighborhood. For ray tracing, we shoot one ray per pixel into the scene (allowing a single reflection from the object). A single shadow-ray then determines whether or not a pixel is lit by a point light source. Whereas, to compute the photon map, a large number of photons will be emitted from a light source. For each photon bounce, we will compute a single corresponding shadow photon. Photon mapping is enabled by replacing the final lighting step of ray tracing with a photon gathering step. For any scene point being rendered, photons are integrated over a fixed-size area.

Expected Result:

We would use photon mapping to realistically simulate the interaction of light with objects in the scene. With increase in the number of photons, the accuracy of the final rendered image will also increase. We will try to simulate a Cornell Box scene consisting of a cube bounding volume, one spherical object and one cube object. We will color the cube walls in different colors to understand the effects of reflection. We will keep a light source located on the top face of the bounding cube.